What is Linux and why is it so popular?

Manjunath.R

#16/1, 8th Main Road, Shivanagar, Rajajinagar, Bangalore560010, Karnataka, India

*Corresponding Author Email: manjunath5496@gmail.com

*Website: http://www.myw3schools.com/

Abstract

Whether you know it or not you are already using Linux (the best-known and most-used open source operating system) every day. From supercomputers to smartphones, the Linux operating system is everywhere. As an operating system, Linux is a family of open source Unix-like software based on the Linux kernel - that sits underneath all of the other software on a computer, receiving requests from those programs and relaying these requests to the computer's hardware. With regard to careers, it is becoming increasingly valuable to have Linux skills rather than just knowing how to use Windows. In general, Linux is harder to manage than Windows, but offers more flexibility and configuration options.

Linus Benedict Torvalds is a Finnish-American software engineer who is the creator and, historically, the main developer of the Linux kernel, used by Linux distributions and other operating systems such as Android and Chrome OS.

"Linux has definitely made a lot of sense even in a purely materialistic sense."

− Linus Torvalds

Introduction

Every desktop computer uses an operating system. The most popular operating systems in use today are: Windows, Mac OS, and **LINUX**. Linux is the best-known notoriously reliable and highly secure open source portable operating system -- very much like UNIX -- that has become very popular over the last several years -- created as a task done for pleasure by **Linus Torvalds** -- computer science student at the University of Helsinki in Finland -- in the early 1990s and later developed by more than a thousand people around the world.

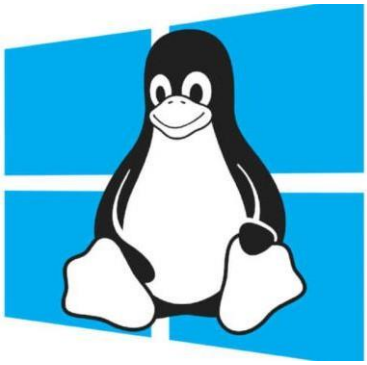
Linux is fast, free and easy to use, that sits underneath all the other software on a computer -- runs your computer -- handling all interactions between you and the hardware i.e., whether you're typing a letter, calculating a money budget, or managing your food recipes on your computer, the Linux operating system (similar to other Operating Systems, such as **Windows XP**, Windows 7, Windows 8, and Mac OS X) provides the essential air that your computer breathes.

Linux is the most important technology advancement of the twenty-first century and Licensed under the General Public License (GPL) that Linux uses ensures that the software will always be open to anyone and whose source code is open and available for any user to check, which makes it easier to find and repair vulnerabilities and it power the laptops, development machines and servers at Google, Facebook, Twitter, NASA, and New York Stock Exchange, just to name a few. Linux has many more features to amaze its users such as: Live CD/USB, **Graphical user interface** (X Window System) etc.

Why LINUX?

Although Microsoft Windows (which is the most likely the victim of viruses and malware) has made great improvements in reliability in recent years, it considered less reliable than Linux. Linux is notoriously reliable and secure and it is free from constant battling **viruses** and **malware** (which may

affect your desktops, laptops, and servers by corrupting files, causing slow downs, crashes, costly repairs and taking over basic functions of your **operating system**) – and it keep yourself free from licensing fees i.e., zero cost of entry ... as in free. You can install Linux on as many reliable computer ecosystems on the planet as you like without paying a cent for software or server licensing. While **Microsoft Windows** usually costs between $99.00 and $199.00 USD for each licensed copy and fear of losing data.



Below are some examples of where Linux is being used today:

- Android phones and tablets
- Servers
- TV, Cameras, DVD players, etc.
- Amazon
- Google
- U.S. Postal service
- New York Stock Exchange

Linux Operating System has primarily three components:

1. **Kernel**

Kernel is the core part of Linux Operating System and interacts directly with hardware. It is responsible for all major activities of the Linux operating system.

2. **System Library**

System libraries are special programs using which application programs accesses Kernel's features.

3. **System Utility**

System Utility programs are responsible to do specialized tasks.

Important features of Linux Operating System:

- Portable
- Open Source
- Multi-User
- Multiprogramming
- Hierarchical File System
- Security

Now Linux (successfully being used by several millions of users worldwide) has grown passed the stage where it was almost exclusively an academic system, useful only to a handful of people with a technical background. It provides more than the operating system: there is an entire infrastructure supporting the chain of effort of creating an operating system, of making and **testing programs** for it, of bringing everything to the users, of supplying maintenance, updates and support and customizations, runs on different platforms including the Intel and **Alpha platform**. Today, Linux is ready to accept the challenge of a fast-changing world to do various types of operations, call application programs etc. Since the hiring focus is shifting more and more toward **DevOps** type skills, a Linux skill set will be the types of things that will make you very deployable.

Paul Buchheit: Then you have what we do with PCs, and that's technically pretty challenging—to take this big network of machines that are unreliable and build a big, reliable storage system out of it.

A lot of the machines that Google is built on—commodity is the polite word for them—they're regular PCs and so they're not always the most reliable.

— Jessica Livingston

Linux Commands

The command-line interface is one of the nearly all well built trademarks of Linux. There exists an ocean of Linux commands, permitting you to do nearly everything you can be under the impression of doing on your Linux operating system. Although, this to the end of time creates a problem: by all of so copious commands accessible to manage, you don't comprehend where and at which point to fly learning them, especially when you are learner. If you are facing this problem, and are peering for a painless method to begin your command line journey in Linux, you've come to the right place, we will launch you to a hold of well liked and helpful Linux commands.

Description:

Display system date and time.

Command:

```
date
```

Description:

Display calendar.

Command:

```
cal
```

Description:

Display date, time and calendar.

Command:

```
date & cal
```

Description:

Display August month 2016 year calendar.

Command:

```
cal 8 2016
```

Description:

Used to clear the terminal window.

Command:

```
clear
```

Description:

Exit from the terminal window.

Command:

```
exit
```

Description:

Display free and used system memory.

Command:

```
free
```

Description:

Display free and used system memory in bytes.

Command:

```
free  -b
```

Description:

Display free and used system memory in kilobytes.

Command:

```
free  -k
```

Description:

Display free and used system memory in megabytes.

Command:

```
free  -m
```

Description:

Change user password.

Command:

```
passwd
```

Description:

Power-off the machine.

Command:

```
shutdown
```

Description:

Power-off the machine immediately.

Command:

```
shutdown  -h now
```

Description:

Power-off the machine after 10 minutes.

Command:

```
shutdown  -h +10
```

Description:

Print current working directory.

Command:

```
echo $PWD
```

Description:

Print previous working directory.

Command:

```
echo $OLDPWD
```

Description:

Executes the 11th command in command history.

Command:

```
!11
```

Description:

Reveals your command history.

Command:

```
history
```

Description:

Power off or reboot the Operating system.

Command:

```
sudo reboot
```

Description:

Display the IP address of the host.

Command:

```
ip address
```

Description:

List the size of files and directories.

Command:

```
ls  -s
```

Description:

View mounted file systems.

Command:

```
mount
```

Description:

Display the information of disk usage of files and directories.

Command:

```
du
```

Description:

Tells you how long the system has been running.

Command:

```
uptime
```

Description:

Set current date as 02 Nov 1988.

Command:

```
date -- set 1998-11-02
```

Description:

Set current time as 12:11:02 IST.

Command:

```
date -- set 12:11:02
```

Description:

View Specific Disk Partition in Linux.

Command:

```
fdisk -l /dev/sda
```

Description:

Lists all files and directories in the present working directory.

Command:

```
ls
```

Description:

Report the process information.

Command:

```
ps
```

Description:

Display disk usage.

Command:

```
df
```

Description:

Display disk usage in gigabytes, megabytes, or kilobytes.

Command:

```
df  -H
```

Description:

Delete every file and every directory.

Command:

```
rm  -r *
```

Description:

Provides a quick overview of the currently running processes.

Command:

```
top
```

Description:

The system performs an immediate reboot.

Command:

```
reboot
```

Description:

Terminate processes without having to log out or reboot.

Command:

```
kill
```

Description:

Change the current working directory.

Command:

```
cd
```

Description:

Create a new session on the system.

Command:

```
login
```

Description:

List open files.

Command:

```
lsof
```

Description:

List USB devices.

Command:

```
lsusb
```

Description:

Check the status of the network services.

Command:

```
service network status
```

Description:

Start the network service.

Command:

```
service network start
```

Description:

Stop the network service.

Command:

```
service network stop
```

Description:

Restart the network service.

Command:

```
service network restart
```

Description:

Report information about the users currently on the machine and their processes.

Command:

```
w
```

Description:

Display the current directory.

Command:

```
pwd
```

Description:

Displays CPU architecture information (such as number of CPUs, threads, cores, sockets, and more).

Command:

```
lscpu
```

Description:

Displays the number of processing units available to the current process.

Command:

```
nproc
```

Description:

The system performs an immediate reboot.

Command:

```
init 6
```

Description:

Power-off the machine.

Command:

```
init 0
```

Description:

List files by date.

Command:

```
ls -lrt
```

Description:

Report information about storage devices such as hard disks, flash drives etc.

Command:

```
lsblk
```

Description:

Show exit status of previous command.

Command:

```
echo $?
```

Description:

Lists a few useful info commands.

Command:

```
info
```

Description:

Prints current year's calendar.

Command:

```
cal -y
```

Description:

Check the status of all the services.

Command:

```
service --status-all
```

Description:

Display time in hh:mm:ss.

Command:

```
date +%T
```

Description:

Tells when the user last logged on and off and from where.

Command:

```
last  -1 username
```

Description:

Sort files and directories by extension name.

Command:

```
ls  -X
```

Description:

Display the manual for the pwd command.

Command:

```
man pwd
```

Description:

Displays information about running processes in the form of a tree.

Command:

```
pstree
```

Description:

Resets your terminal.

Command:

```
reset
```

Description:

Displays What date is it this Friday.

Command:

```
date -d fri
```

Description:

Displays the size of each individual file.

Command:

```
du  -a
```

Description:

Display information about the Advanced configuration and power Interface.

Command:

```
acpi
```

Description:

Takes you two folders back.

Command:

```
cd ../..
```

Description:

Takes you to the previous directory.

Command:

```
cd -
```

Description:

Displays a list of shell built-in commands.

Command:

```
help
```

Description:

Lists your last logins.

Command:

```
last yourusername
```

Description:

Create a new directory called myfiles.

Command:

```
mkdir myfiles
```

Description:

Remove the directory myfiles.

Command:

```
rmdir myfiles
```

Description:

Disable password for a specific user "root1".

Command:

```
passwd -d root1
```

Description:

Switch to user "root1".

Command:

```
sudo su root1
```

Description:

Exit from the terminal window.

Command:

```
logout
```

Description:

Creates a user "root1".

Command:

```
useradd "root1"
```

Description:

Assign password to user "root1".

Command:

```
passwd "root1"
```

Description:

Repeats the last command.

Command:

```
!!
```

Description:

Display Who you are logged in as.

Command:

```
whoami
```

Description:

Display the login name of the current user.

Command:

```
logname
```

Description:

Report the name of the kernel.

Command:

```
uname
```

Description:

Print the kernel version.

Command:

```
uname  -v
```

Description:

Print the operating system.

Command:

```
uname  -o
```

Description:

Report the machine hardware name.

Command:

```
uname  -m
```

Description:

Print version information and exit.

Command:

```
uname  --version
```

Description:

Print the kernel release.

Command:

```
uname  -r
```

Description:

Report the network node hostname.

Command:

```
uname  -n
```

Description:

Display all port connections (both TCP and UDP).

Command:

```
netstat  -a
```

Description:

Display only TCP (Transmission Control Protocol) port connections.

Command:

```
netstat  -at
```

Description:

Display only UDP (User Datagram Protocol) port connections.

Command:

```
netstat  -au
```

Description:

Display all active listening ports.

Command:

```
netstat  -I
```

Description:

Display all active listening TCP ports.

Command:

```
netstat  -It
```

Description:

Display all active listening UDP ports.

Command:

```
netstat  -lu
```

Description:

Reveal all the information about the current user (user id, username, group id, group name etc.).

Command:

```
id
```

Description:

Reveal all the information about the user "root1" (user id, username, group id, group name etc.).

Command:

```
id  root1
```

Description:

Print the machine's architecture.

Command:

```
arch
```

Description:

Display the list of available fonts.

Command:

```
fc-list
```

Description:

Create two directories (myfiles, files).

Command:

```
mkdir myfiles files
```

Description:

install apache (CentOS).

Command:

```
yum install httpd
```

Description:

install apache (Ubuntu).

Command:

```
apt install httpd
```

Description:

upgrade apache (CentOS).

Command:

```
yum update httpd
```

Description:

upgrade apache (Ubuntu).

Command:

```
apt update httpd
```

Description:

uninstall apache (CentOS).

Command:

```
yum remove httpd
```

Description:

uninstall apache (Ubuntu).

Command:

```
apt remove httpd
```

Description:

Display usage summary for the command (date).

Command:

```
date --help
```

Description:

List active connections to/from system.

Command:

```
ss -tup
```

Description:

List internet services on a system.

Command:

```
ss -tupl
```

Description:

Display all active UNIX listening ports.

Command:

```
netstat  -lx
```

Description:

Display all the active interfaces details.

Command:

```
ifconfig
```

Description:

Display information of all network interfaces.

Command:

```
ifconfig  -a
```

Description:

Compare the contents of two files (1.txt, 2.txt).

Command:

```
diff 1.txt 2.txt
```

Description:

Tells you how many lines, words, and characters there are in a file (1.txt).

Command:

```
wc 1.txt
```

Description:

Compresses file (1.txt), so that it take up much less space.

Command:

```
gzip 1.txt
```

Description:

Uncompresses file (1.txt) compressed by gzip.

Command:

```
gunzip 1.txt
```

Description:

Examine the contents of the file (1.txt).

Command:

```
cat 1.txt
```

Description:

Display calendar.

Command:

```
ncal
```

Description:

Removes the file (1.txt).

Command:

```
rm 1.txt
```

Description:

Rename a file named 1.txt to 0.txt.

Command:

```
mv 1.txt 0.txt
```

Description:

Replace the contents of 0.txt with that of 1.txt.

Command:

```
cp 1.txt 0.txt
```

Description:

Create a empty file (test.txt).

Command:

```
touch test.txt
```

Description:

Print the last 10 lines of a file (1.txt).

Command:

```
tail 1.txt
```

Description:

Print N number of lines from the file (1.txt).

Command:

```
tail  -n N 1.txt
```

Description:

Prints the number of words in a file (1.txt).

Command:

```
wc -w 1.txt
```

Description:

Prints the number of characters from a file (1.txt).

Command:

```
wc -m 1.txt
```

Description:

Prints the length of the longest line in a file (1.txt).

Command:

```
wc -L 1.txt
```

Description:

Print information about usb ports, graphics cards, network adapters etc.

Command:

```
lspci
```

Description:

View contents of a file (1.txt).

Command:

```
less 1.txt
```

Description:

Display calendar (last month, current month, and next month).

Command:

```
cal -3
```

Description:

Compare the contents of three files (1.txt, 2.txt, 3.txt) line by line.

Command:

```
diff3 1.txt 2.txt 3.txt
```

Description:

Compare two files (1.txt, 2.txt) line-by-line.

Command:

```
comm 1.txt 2.txt
```

Description:

Perform byte-by-byte comparison of two files (1.txt, 2.txt).

Command:

```
cmp 1.txt 2.txt
```

Description:

Prints the CRC checksum and byte count for the file "myfiles.txt".

Command:

```
cksum myfiles.txt
```

Description:

Append contents of files (1.txt, 2.txt) into one file (0.txt).

Command:

```
cat 1.txt 2.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  r 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  h 1.txt 2.txt 3.txt > 0.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt).

Command:

```
sed  -n  p 1.txt 2.txt 3.txt > 0.txt
```

Shortcuts:

```
| ctrl+c           | Halts the current command           |
| ctrl+z           | Stops the current command           |
|                  |                                     |
| ctrl+d           | Logout the current session          |
| ctrl+w           | Erases one word in the current line |
|                  |                                     |
| ctrl+u           | Erases the whole line               |
| ctrl+r           | Type to bring up a recent command   |
```

Description:

Writes contents of a file (0.txt) to output, and prepends each line with line number.

Command:

```
nl 0.txt
```

Description:

Create a empty file (test1.txt) inside a directory (test).

Command:

```
mkdir test
cd test
pwd
touch test1.txt
```

Description:

Gather information about hardware components such as CPU, disks, memory, USB controllers etc.

Command:

```
sudo lshw
```

Description:

Gather information about file system partitions.

Command:

```
sudo fdisk -l
```

Description:

Displays the line (good morning) in which the string (good) is found in the file (1.txt).

Command:

```
grep good 1.txt
```

Description:

Append contents of files (1.txt, 2.txt, 3.txt) into one file (0.txt) using for loop.

Command:

```
for i in {1..3}; do cat "$i.txt" >> 0.txt; done
```

Description:

Search for files (test.txt, test1.txt, test2.txt, test.php, test.html) in a directory as well as its sub-directories.

Command:

```
find test*
```

Description:

Displays status related to a file (1.txt).

Command:

```
stat 1.txt
```

```
###
|     Command       |  Description        |
|:------------------|-----------------:   |
| vi                | Open vi editor      |
| i                 | Go to Insert mode   |
|                   |                     |
| a =20; b =64;     |                     |
| print (a + b);    |                     |
| Hit Escape to return to Normal mode.  |
| :w hello.py       | Save text           |
| :q                | Quit                |
| python hello.py   |Print the output:84  |
```

Description:

Download the file (file.txt) from url "http: //website.com/files/file.txt".

Command:

```
wget http://website.com/files/file.txt
```

Description:

Display host's numeric ID in hexadecimal format.

Command:

```
hostid
```

Description:

Display file type of the file (myfiles.txt).

Command:

```
file myfiles.txt
```

Description:

Create a file (myfiles.txt) containing a text (Hello World).

Command:

```
echo 'Hello World' > myfiles.txt
```

Description:

Create a file (myfiles.txt) containing a text (Hello World).

Command:

```
printf 'Hello World' > myfiles.txt
```

Description:

Display IP address of the hostname.

Command:

```
hostname  -i
```

Description:

Add a new line of text to an existing file (1.txt).

Command:

```
echo "Hello world!" >> 1.txt
echo "this is 2nd line text" >> 1.txt
echo "last line!" >> 1.txt
```

Description:

Displays a single line description about a command (cal).

Command:

```
whatis cal
```

```
###
|     Command      |  Description       |
|:---------------------|-----------------:      |
| vi                | Open vi editor     |
| i                 | Go to Insert mode  |
| Type some text.   |                    |
| Hit Escape to return to Normal mode.  |
| :w test.txt       | Save text          |
| :q                | Quit               |
| :q!               |Quit without saving |


###
|     Command      |  Description          |
|:---------------------|-----------------:         |
| vi                | Open vi editor        |
| i                 | Go to Insert mode     |
| $name = "Paul";   |                       |
| print "$name";    |                       |
| Hit Escape to return to Normal mode.     |
| :w hello.pl       | Save text             |
| :q                | Quit                  |
| perl hello.pl     |Print the output: Paul |


###
|     Command                 |  Description           |
|:-----------------------------------|-----------------:          |
| vi                          | Open vi editor         |
| i                           | Go to Insert mode      |
| echo "What is your name?"   |                        |
| read PERSON                 |                        |
| echo "Hello, $PERSON"       |                        |
| Hit Escape to return to Normal mode.                 |
| :w hello.sh                 | Save text              |
| :q                          | Quit                   |
| sh hello.sh                 | Output:                |
|                             | What is your name?     |
|                             | If you enter: Zara Ali |
|                             | Hello, Zara Ali        |
```

Description:

Check the network connectivity between host (your connection) and server (Google server).

Command:

```
ping google.com
```

Description:

Find the location of source/binary file of a command (cal).

Command:

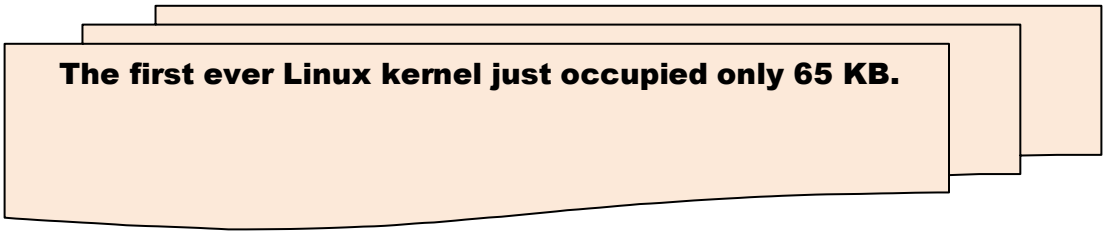
```
whereis cal
```

There are 2 ways to use the command:

- Numeric mode
- Symbolic mode

Numeric mode	Permission Type	Symbolic mode
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x
6	Read + Write	rw-
7	Read + Write + Execute	rwx

90% of the public cloud workload is run on Linux distros.



import subprocess subprocess.call ('**linux command**')	import os os.system('**linux command**')

All of the 500 fastest supercomputers run Linux.

import os os.system('**ls**')	**List all the files and directories in the current directory**
import subprocess subprocess.call ('**ls**')	

Execution of the linux command "ls" using the python program

Command:

```
last reboot
```

Description:

Show system reboot history

Command:

```
dmesg
```

Description:

Displays the messages from the kernel ring buffer (a data structure that records messages related to the operation of the kernel**)**

Command:

```
cat /proc/cpuinfo
```

Description:

Display CPU information

Command:

```
cat /proc/meminfo
```

Description:

Display memory information

Command:

```
lspci -tv
```

Description:

Display PCI (Peripheral Component Interconnect**) devices**

Command:

```
lsusb –tv
```

Description:

Display USB devices

Command:

```
free -h
```

Description:

Display free and used memory (-h for human readable, -m for MB, -g for GB**)**

Command:

```
mpstat 1
```

Description:

Display processor related statistics

Command:

```
vmstat 1
```

Description:

Display virtual memory statistics

Command:

```
iostat 1
```

Description:

Display Input / Output statistics

Command:

```
watch df -h
```

Description:

Execute "df -h" command, showing periodic updates

Command:

```
ps –ef
```

Description:

Display all the currently running processes on the system

Command:

```
ip a
```

Description:

Display all network interfaces and IP address

Command:

```
dig wikipedia.org
```

Description:

Display DNS information for domain (wikipedia.org**)**

Command:

```
host wikipedia.org
```

Description:

Display the IP address details of the specified domain (wikipedia.org**)**

Command:

```
netstat -nutlp
```

Description:

Display listening Transmission Control Protocol (TCP**) and the User Datagram Protocol (**UDP**) ports and corresponding programs**

Command:

```
rpm -qa
```

Description:

List all installed packages

Command:

```
yum list installed
```

Description:

List all installed packages (CentOS)

Command:

```
yum info httpd
```

Description:

Display description and summary information about package "httpd" **(CentOS)**

Command:

```
du -ah
```

Description:

Display disk usage for all files and directories in human readable format

Command:

```
du -sh
```

Description:

Display total disk usage off the current directory

Command:

```
cd /etc
```

Description:

Change to the /etc directory

Command:

```
ps -A
```

Description:

List the status of all the processes along with process id and PID

Command:

```
#include <stdio.h>
int main()
{
  printf("Hello world\n");
  return 0;
}
```

Hello.c

```
gcc Hello.c
```

Description:

Compile the C program saved in Hello.c file

Command:

```
#include <iostream>
int main()
{
std::cout << "Hello world!";
  return 0;
}
```

Hello.cpp

```
g++ Hello.cpp
```

Description:

Compile the C++ program saved in Hello.cpp file

Command:

```
tty
```

Description:

Displays the file name of the terminal connected to standard input

Command:

```
public class MyClass {
public static void main(String [] args) {
System.out.println("Hello, World!");
}
}

javac MyClass.java
```

MyClass.java

Description:

Compile the Java program saved in MyClass.java file using javac compiler

Command:

```
od -b myfiles.txt
```

Description:

Displays the contents of myfiles.txt **file in octal format**

Command:

```
od -c myfiles.txt
```

Description:

Displays the contents of myfiles.txt **file in character format**

Command:

```
od -An -c myfiles.txt
```

Description:

Displays the contents of myfiles.txt **file in character format but with no offset information**

Command:

```
csplit myfiles.txt 13 62 101
```

Description:

If the file **myfiles.txt** has 123 lines, the **csplit command** would create four files: the xx00 file would contain lines 1−12, the xx01 file would contain lines 13−61, the xx02 file would contain lines 62−100, the xx03 file would contain lines 101−123

Command:

```
md5sum myfiles.txt
```

Description:

Prints a 32-character (128-bit) checksum of myfiles.txt **file using the MD5 algorithm**

Command:

```
more myfiles.txt
```

Description:

Displays the content of myfiles.txt **file**

Command:

```
sha1sum myfiles.txt
```

Description:

Prints SHA1 (160-bit) checksum of myfiles.txt **file**

SHA 1 → Secure Hash Algorithm 1

Command:

```
shred myfiles.txt
```

Description:

Overwrites the myfiles.txt **file repeatedly − in order to make it harder for even very expensive hardware probing to recover the data**

Command:

```
cat myfile.txt

01. Einstein
02. Newton
03. Maxwell
04. Tesla
05. Edison

tac myfile.txt

05. Edison
04. Tesla
03. Maxwell
02. Newton
01. Einstein
```

Description:

Print the lines of myfile.txt **in reverse (from last line to first)**

Command:

```
uniq myfiles.txt
```

Description:

Delete repeated lines in the file (myfiles.txt).

Command:

```
chkconfig --list
```

Description:

Displays a list of system services and whether they are started (on**) or stopped (**off**) in run levels 0−6**

Command:

```
halt -p
```

Description:

Power-off the system

Command:

```
xdg-open myfiles.txt
```

Description:

Open a file (myfiles.txt).

Command:

```
lastlog
```

Description:

Prints the details of the last login (login-name, port and last login time**)**

Command:

```
lastlog -t 1
```

Description:

Displays the login information (1 day ago**)**

Command:

```
lastlog -u manju
```

Description:

Display lastlog information for a particular user (manju**)**

Command:

```
cat /etc/passwd
```

```
more /etc/passwd
```

```
less /etc/passwd
```

```
getent passwd
```

Description:

List all users on Linux

Command:

```
tail -5 /etc/passwd

head -5 /etc/passwd
```

Description:

List last 5 users on Linux

List first 5 users on Linux

Command:

```
wall "The system will be shutdown in 10 minutes."
```

Description:

The message (The system will be shutdown in 10 minutes.**) will be broadcasted to all users that are currently logged in**

Command:

```
chage -l manju
```

Description:

List the password and its related details for a user (manju**)**

Command:

```
chage -M 10 manju
```

Description:

Set Password Expiry Date for an user (manju**)**

Command:

```
chage -E "2020-07-30" manju
```

Description:

Set the Account Expiry Date for an User (manju**)**

Command:

```
chage -I 10 manju
```

Description:

Force the user (manju**) account to be locked after 10 inactivity days**

Command:

```
ftp 192.168.42.77
```

Description:

Connect to an FTP server at remote server IP address "192.168.42.77**"**

Command:

```
arp –a
```

Description:

Lists all the peers connected at various interfaces along with their MAC Addresses **and IP addresses**

Command:

```
dnsdomainname
```

Description:

Display the system's DNS domain name

Command:

```
domainname
```

Description:

Display the name of the domain your machine belongs to

Command:

```
echo 'Hello World!' | base64

Output: SGVsbG8gV29ybGQhCg==
```

Description:

Encode text (Hello World!**) to base64**

Command:

```
echo 'SGVsbG8gV29ybGQhCg==' | base64 -d

Output: Hello World!
```

Description:

Decode (SGVsbG8gV29ybGQhCg==**) to text (**Hello World!**)**

Command:

```
fc-cache -f –v
```

Description:

Build font information cache files

Command:

```
cat 1.txt

Einstein
Newton
Albert

fmt 1.txt

Einstein Newton Albert
```

Description:

Formats text in a single line

```
cat phy.txt

Albert Einstein was a German-born theoretical physicist, widely acknowledged to
be one of the greatest physicists of all time. Einstein is known for developing
the theory of relativity, but he also made important contributions to the
development of the theory of quantum mechanics.


fmt -w 1  phy.txt

 Albert
 Einstein
 was
 a
 German-born
 theoretical
 physicist,
 widely
 acknowledged
 to
 be
 one
 of
 the
 greatest
 physicists
 of
 all
 time.
 Einstein
 is
 known
 for
 developing
```

```
 the
 theory
 of
 relativity,
 but
 he
 also
 made
 important
 contributions
 to
 the
 development
 of
 the
 theory
 of
 quantum
 mechanics.
```

```
cat phy.txt

Albert Einstein was a German-born theoretical physicist, widely acknowledged to
be one of the greatest physicists of all time. Einstein is known for developing
the theory of relativity, but he also made important contributions to the
development of the theory of quantum mechanics.



fold -w 20  phy.txt

Albert Einstein was
```

```
 a German-born theor
etical physicist, wi
dely acknowledged to
 be one of the great
est physicists of al
l time. Einstein is
known for developing
 the theory of relat
ivity, but he also m
ade important contri
butions to the devel
opment of the theory
 of quantum mechanic
s.
```

Command:

```
traceroute google.com
```

Description:

Prints the route that a packet takes to reach the Google (172.217.26.206**) host from the local machine**

Command:

```
cat 1.txt
```

```
Einstein
Newton
Albert

gzip 1.txt

zcat 1.txt.gz

Einstein
Newton
Albert
```

Description:

View the contents of zipped file

Command:

```
zdiff 1.txt.gz 2.txt.gz
```

Description:

Compare the contents of two zipped files (1.txt.gz, 2.txt.gz**)**

Command:

```
ss | less
```

Description:

List all connections

Command:

```
ss -aA tcp
```

Description:

Filter out TCP (Transmission Control Protocol**) connections**

Command:

```
ss -aA udp
```

Description:

Filter out UDP (User Datagram Protocol**) connections**

Command:

```
ss -lnt
```

Description:

Display only listening sockets

Command:

```
ss -ltp
```

Description:

Print process name and PID

Command:

```
ss -s
```

Description:

Print summary statistics

Command:

```
ss -tl6
```

Description:

Display only IPv6 connections

Command:

```
ss -tl -f inet
```

Description:

Display only IPv4 socket connections

Command:

```
ss -t4 state established
```

Description:

Display all IPv4 TCP sockets that are in connected state

Command:

```
pmap 3244
```

Description:

View the memory map of a process with Process ID (3244**)**

Command:

```
apropos -r 'remove file'
```

Description:

Find command that removes file

Command:

```
apropos editor
```

Description:

Display information about the editing programs that are available on a system

Command:

```
apropos pstree
```

Description:

Provide information about the pstree command (which displays the names of the processes currently on the system in the form of a tree diagram**)**

The **apropos command** is useful when you know what you want to do, but you have no idea what command you should be using to do it. If you were wondering how to locate files, for example, the commands

```
apropos find
```

and

```
apropos locate
```

would have a lot of suggestions to offer.

```
basename /etc/passwd

Output: passwd

basename /usr/local/apache2/conf/httpd.conf
```

```
Output: httpd.conf
```

```
echo a b c d e f| xargs

Output: a b c d e f


echo a b c d e f| xargs -n 3

Output:

a b c
d e f
```

display only 3 items per line

Command:

```
env
```

Description:

Print out a list of all environment variables

Command:

```
printenv HOME
```

Description:

Print HOME variable value

```
cat score.txt

Albert-30
John-50
William-80
Stephen-20
Justin-40

cut -d- -f2 score.txt

30
50
80
20
40

cut -d- -f1 score.txt

Albert
John
William
Stephen
Justin
```

```
cat 1.txt

Hello World


cat 2.txt

Computer Program


paste 1.txt  2.txt

Hello World   Computer Program
```

```
cat 1.txt

Hello World


cat 2.txt

Computer Program


join 1.txt  2.txt

Hello World   Computer Program
```

Command:

```
rev 1.txt
```

Description:

Reverse lines of a file (1.txt**)**

```
cat 3.txt

22
33
11
77
55

sort 3.txt

11
22
33
55
77
```

sorts numeric values in 3.txt file and displays sorted output

```
cat 1.txt

Hello World


cat 1.txt | tr "[a-z]" "[A-Z]"

HELLO WORLD
```

convert from lower case to upper case

```
cat 5.txt

zz
zz
yy
yy
yy
xx

uniq 5.txt

zz
yy
xx
```

removes duplicate lines and displays unique lines

```
cat 6.txt

Einstein
Newton
Tesla


nl 6.txt

1 Einstein
2 Newton
3 Tesla
```

numbers the lines in a file (6.txt)

Command:

```
ls -l *.txt
```

Description:

Lists the files with .txt **extension**

> The thing with Linux is that the developers themselves are actually customers too: that has always been an important part of Linux.
>
> **Linus Torvalds**

Linux	Unix
Free to use (**open source**)	Licensed Operating System (**closed source**)
Linux is just the kernel	Unix is a complete package of Operating System
Bash (**Bourne Again SHell**) is default shell for Linux	Bourne Shell is default shell for Unix
Portable and is booted from a USB Stick	Unportable
Source code is accessible to the general public	Source code is not accessible to anyone
Uses Graphical User Interface with an optional Command Line Interface	Uses Command Line Interface

Command:

```
echo $SHELL
```

Description:

Print the Default shell of user

Command:

```
echo $0
```

Description:

Display the name of the currently running process **($0 is the name of the running process**). If you use it inside of a shell then it will return the name of the shell. If you use it inside of a script, it will return the name of the script

Command:

```
echo *
```

Description:

Print all files and folders − similar to ls command

Command:

```
ps -p $$

Output:

   PID  TTY          TIME   CMD
  3352 pts/0    00:00:00 bash
```

Description:

Print the process ID of the current shell ($$ is the process ID of the current shell)

Command:

```
cat /etc/shells
```

Description:

List shells

Command:

```
echo m*
```

Description:

Display the files in the current folder that start with the letter "m".

Command:

```
last
```

Description:

List last logins of users and what happened such as "shutdown**" or "**crash**" etc.**

Command:

Command:

```
echo ~
```

Description:

Print your home folder path

```
bzip2 -k phy.txt
```

Description:

Compresses but does not deletes the original file

phy.txt → phy.txt.bz2

Command:

```
bzip2 -d phy.txt.bz2
```

Description:

Decompresses the compressed file (phy.txt.bz2**)**

phy.txt.bz2 → phy.txt

Command:

```
bzcat phy.txt.bz2
```

Description:

Display the contents of compressed file (phy.txt.bz2**)**

Command:

```
bunzip2 phy.txt.bz2
```

Description:

Decompresses the compressed file (phy.txt.bz2**)**

Command:

```
crontab -l
```

Description:

Display current logged-in user's crontab entries

```
cat /dev/null > phy.txt
```

```
cp /dev/null phy.txt
```

```
echo "" > phy.txt
```

```
echo > phy.txt
```

Description:

Empty the content of a file (phy.txt**)**

Command:

```
nohup ping google.com &
```

Description:

Ping google.com **and send the process to the background**

Command:

```
nohup ping google.com > log.txt &
```

Description:

Save the ping logs to log.txt

```
pgrep -a ping

Output:

3858 ping google.com
4200 ping google.com
4236 ping google.com

kill 3858

pgrep -a ping

Output:

4200 ping google.com
4236 ping google.com
```

Command:

```
ls -la /home
```

Description:

Display the contents of /home

Command:

```
sudo shutdown 2
```

Description:

Power-off the machine after 2 minutes

Command:

```
shutdown -c
```

Description:

Cancel the shutdown process

Command:

```
pr 36.txt
```

Description:

Display the contents of the file (36.txt**) one page after the other**

Command:

```
stty –a
```

Description:

Display all current terminal settings

Command:

```
ls -1
```

Description:

List files one per line

Command:

```
yes John
```

Description:

Outputs a string (John**) repeatedly until killed**

Command:

```
vdir
```

Description:

List files and directories in the current directory (one per line**) with details**

Command:

```
who -b
```

Description:

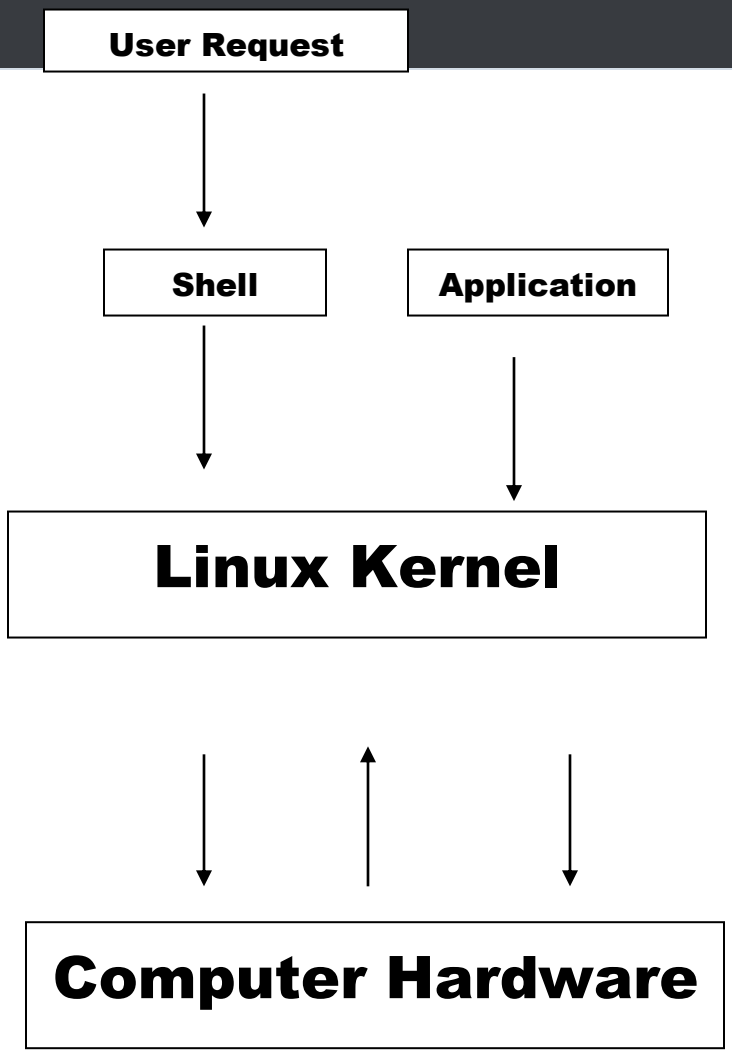
Print when the system was booted

```
# Open phy.txt with nano
```

```
nano phy.txt


# Open phy.txt with vim

vim phy.txt
```




```
w --ip-addr
```

```
# Displays information regarding the users currently on the machine, login time, IDLE time,
TTY and CPU time

Output:

11:12:10 up  1:29,  2 users,  load average: 0.02, 0.04, 0.10
USER     TTY      FROM  LOGIN@   IDLE   JCPU   PCPU WHAT
manju    :0       :0    02:43   ?xdm?   3:30   0.65s gdm-session-worker [pa
manju    pts/0    :0    11:01    2.00s  0.10s  0.01s w --ip-addr

w –short
# Omits CPU time and login information
Output:

11:11:46 up  1:28,  2 users,  load average: 0.02, 0.04, 0.11
USER     TTY      FROM              IDLE WHAT
manju    :0       :0               ?xdm?  gdm-session-worker [pam/gdm-password]
manju    pts/0    :0                2.00s w --short
```

Command:

```
findmnt
```

Description:

Display a list of currently mounted file systems

Command:

```
ip addr show
```

Description:

List IP addresses and network interfaces

Command:

```
netstat -pnltu
```

Description:

List active (listening) ports

Command:

```
Journalctl
```

Description:

Display systemd, kernel and journal logs

Command:

```
sudo systemctl status network
```

Description:

Display the status of network service

Command:

```
sudo systemctl start network
```

Description:

Start the network service

Command:

```
sudo systemctl stop network
```

Description:

Stop the network service

Command:

```
sestatus -b
```

Description:

Display the current state of Booleans

Command:

```
getenforce
```

Description:

Reports whether SELinux **is enforcing, permissive or disabled**

Security-Enhanced Linux (SELinux) is a security architecture for Linux systems that allows administrators to have more control over who can access the system

```
setenforce 0
```

```
getenforce
```

```
Output:

Permissive
```

```
setenforce 1
```

```
getenforce
```

```
Output:

Enforcing
```

- **Enforcing** - SELinux security policy is enforced.
- **Permissive** - SELinux prints warnings instead of enforcing.
- **Disabled** - No SELinux policy is loaded.

Command:

```
sestatus
```

Description:

Display the current status of the SELinux that is running on your system

Command:

```
ps –aef
```

Description:

Display full listing of processes on your system

Command:

```
sar
```

Description:

Display System Activity Report

Command:

```
ulimit
```

Description:

Report the resource limit of the current user

```
Output:

Unlimited          The current user can consume all the resources the current system supports
```

2 types of resource limitation:

- **Hard resource limit:** The physical limit that the user can reach.
- **Soft resource limit:** The limit that is manageable by the user (**its value can go up to the hard limit**)

Command:

```
ulimit -a
```

Description:

Report all the resource limits for the current user

Command:

```
ulimit –s
```

Description:

Check the maximum stack size of the current user

Command:

```
ulimit –e
```

Description:

Check out the max scheduling priority of the current user

Command:

```
ulimit -u
```

Description:

Display the maximum number of user processes

Command:

```
ulimit -v
```

Description:

Check out the size of virtual memory

Command:

```
ulimit -n
```

Description:

Check out how many file descriptors a process can have

Command:

```
man limits.conf
```

Description:

Display the in-depth information on the limits.conf **configuration file**

Command:

```
sar -V
```

Description:

Display the sar version

Command:

```
sar -u 2 5
```

Description:

Report CPU details total 5 times with the interval of 2 seconds

Command:

```
sar -n DEV 1 3 | egrep -v lo
```

Description:

Report about network interface, network speed, IPV4, TCPV4, ICMPV4 network traffic and errors

Command:

```
sar -v 1 3
```

Description:

Report details about the process, kernel thread, i-node, and the file tables

Command:

```
sar -S 1 3
```

Description:

Report statistics about swapping

Command:

```
sar -b 1 3
```

Description:

Report details about I/O operations like transaction per second, read per second, write per second

Command:

```
sudo systemctl status firewalld
```

Description:

Display the status of the firewalld

Command:

```
sudo systemctl start firewalld
```

Description:

Start the firewalld **service**

firewalld is a firewall management tool for Linux operating systems

Command:

```
firewall-config
```

Description:

Start the graphical firewall configuration tool

firewall-cmd

Command:

```
firewall-cmd --list-all-zones
```

Description:

List all zones

Command:

```
firewall-cmd --get-default-zone
```

Description:

Check the currently set default zone

Command:

```
firewall-cmd --list-services
```

Description:

Display currently allowed service on your system

Command:

```
firewall-cmd --list-ports
```

Description:

List the ports that are open on your system

Command:

```
firewall-cmd --zone=work --list-services
```

Description:

List services that are allowed for the public zone

Command:

```
mtr --report google.com
```

Description:

Provides information about the route that Internet traffic takes between the local system and a remote host (google.com**)**

Command:

```
sudo yum install samba
```

Description:

install Samba (CentOS**)**

Samba is client/server technology that implements network resource sharing across operating systems. With Samba, files and printers can be shared across Windows, Mac and Linux/UNIX clients.

Command:

```
sudo firewall-cmd --add-service samba –permanent
```

Description:

Add Samba service to firewalld

Command:

```
zip q.zip q.txt
```

Description:

Create a zip file (q.zip**)**

Command:

```
unzip q.zip
```

Description:

Unzip a zip file (q.zip**)**

```
zipcloak q.zip

-------------------------------------------------------------------------

# zipcloak prompts you for a password, and then ask you to confirm it:

   Enter password:
   Verify password:

...if the passwords match, it encrypts q.zip file

-------------------------------------------------------------------------


unzip q.zip


#  When you try to unzip the q.zip file, it prompts you for the password before
allowing you to extract the file (q.txt) it contains
```

Command:

```
zgrep -l "Einstein" *
```

Description:

Display the names of the files with the word (Einstein**) present in it**

Command:

```
zipsplit -n 1048576 q.zip
```

Description:

Split q.zip **file to create a sequence of zipfiles** (q1.zip, q2.zip…..) − each no larger than **1048576 bytes** (one megabyte)

```
You could concatenate (q1.zip, q2.zip…..) into a new file, w.zip, with the
command:

                                cat q*.zip > w.zip
```

Command:

```
mtr google.com
```

Description:

Test the route and connection quality of traffic to the destination host **google.com**

Command:

```
route
```

Description:

Display IP routing table of a Linux system

Command:

```
nmcli dev status
```

Description:

View all your network devices

Command:

```
nmcli con show
```

Description:

Check network connections on your system

Command:

```
ss -ta
```

Description:

List all TCP ports (**sockets**) that are open on a server

Command:

```
ss -to
```

Description:

Display all active TCP connections together with their timers

Command:

```
type -a alias
```

Description:

Check Bash Aliases in Linux

```
# Difference between %Y and %y is %Y will print 4 digits while %y will print the last 2 digits of the year.

echo "We are in the year = $(date +%Y)"
echo "We are in the year = $(date +%y)"
```

```
# Difference between %B and %b is, %B will print full month name while %b will print abbreviated month name.

echo "We are in the month = $(date +%b)"
```

```
echo "We are in the month = $(date +%B)"

# Difference between %A and %a is, %A will print full Weekday name while %a will print abbreviated weekday name.

echo "Current Day of the week = $(date +%A)"
echo "Current Day of the week = $(date +%a)"
```

```
echo "Date using %D = $(date +%D)"
echo "Date using %F = $(date +%F)"
```

```
Date using %D = 08/15/21

Date using %F = 2021-08-15
```

```
echo "current time in 24 hour format = $(date +%T)"
```

```
current time in 24 hour format = 01:27:46
```

```
echo "current time in 12 hour format = $(date +%r)"
```

```
current time in 12 hour format = 01:27:47 AM
```

```
# Print yesterday's date and time.
echo "Yesterday = $(date -d "Yesterday")"


# Print Tomorrow date and time.
```

```
echo "tomorrow = $(date -d "tomorrow")"

# Find what is the date and time before 10 days from now.
echo "Before 10 days = $(date -d "tomorrow -10 days")"

# Find last month and next month
echo "Last month = $(date -d "last month" "%B")"
echo "Next month = $(date -d "next month" "%B")"

# Find last year and next year
echo "Last Year = $(date -d "last year" "+%Y")"
echo "Next Year = $(date -d "next year" "+%Y")"
```

Command:

```
ls -lai /
```

Description:

Get the number of inodes of files in a directory (**root directory**)

Command:

```
sudo du --inode /
```

Description:

Get the total number of inodes in the root directory

Command:

```
ss -o state established '( sport = :http or sport = :https )'
```

Description:

Get the list of all clients connected to HTTP (Port 80) or HTTPS (Port 443)

Command:

```
ss -tn src :80 or src :443
```

Description:

List the numerical port numbers

Command:

```
sudo yum install putty
```

Description:

Install **PuTTy** on CentOS

Command:

```
sudo watch netstat -tulpn
```

Description:

Watch TCP and UDP Open Ports in Real-Time

Command:

```
sudo watch ss –tulpn
```

Description:

Watch TCP and UDP Open Ports in Real-Time

Command:

```
timeout 5s ping google.com
```

Description:

Timeout a ping command after 5 seconds

Command:

```
yum install curl
```

Description:

Install curl on CentOS

Command:

```
ss -ua
```

Description:

List all UDP Connections

Command:

```
ss –lu
```

Description:

List all Listening UDP Connections

Command:

```
ss -p
```

Description:

Display the Process IDs related to socket connections

Command:

```
ss -4
```

Description:

Display IPv4 and IPv6 Socket Connections

Command:

```
ss -6
```

Description:

Display IPv6 connections

Command:

```
ss -at '( dport = :22 or sport = :22 )'
```

Description:

Filter Connections by Port Number

"The only way to learn a new programming language is by writing programs in it."

–Dennis Ritchie

Command:

```
sdiff phy.txt score.txt
```

Description:

Show Difference between Two Files (**phy.txt and score.txt**)

Command:

```
history -c
```

Description:

Delete or clear all the entries from bash history

Command:

```
ping -c 5 www.google.com
```

Description:

The ping test will stop after sending 5 packets

```
# count number of lines in each .txt file

ls *.txt | xargs wc -l

# count number of words in each .txt file

ls *.txt | xargs wc -w

# count number of characters in each .txt file

ls *.txt | xargs wc -c

# count lines, words and characters in each .txt file

ls *.txt | xargs wc
```

Command:

```
lslogins -u
```

Description:

Displays user accounts

Command:

```
systemctl list-units --type=service
```

Description:

List all loaded services on your system (whether active; running, exited or failed)

Command:

```
systemctl --type=service
```

Description:

List all loaded services on your system (whether active; running, exited or failed)

Command:

```
systemctl list-units --type=service --state=active
```

Description:

List all loaded but active services

Command:

```
systemctl --type=service --state=active
```

Description:

List all loaded but active services

Command:

```
systemctl list-units --type=service --state=running
```

Description:

List all running services (i.e., all loaded and actively running services)

Command:

```
systemctl --type=service --state=running
```

Description:

List all running services (i.e., all loaded and actively running services)

```
#scan a single port

nc -v -w 2 z 192.168.56.1 22

# scan multiple ports

nc -v -w 2 z 192.168.56.1 22 80

# scan range of ports

nc -v -w 2 z 192.168.56.1 20-25
```

Command:

```
cat /etc/resolv.conf
```

Description:

Find out your DNS Server IP address

Command:

```
less /etc/resolv.conf
```

Description:

Find out your DNS Server IP address

Command:

```
findmnt --poll --mountpoint /mnt/test
```

Description:

Monitor mount, unmount, remount and move actions on a directory (**i.e., on /mnt/test**)

Command:

```
uptime -p
```

Description:

Check Linux Server Uptime

Command:

```
uptime –s
```

Description:

Check Linux Server Starting Time

Command:

```
uptime –h
```

Description:

Display uptime's version information

Command:

```
grep -o -i Justin score.txt | wc –l
```

Description:

Count the number of times "**Justin**" appears in the file (**score.txt**)

Command:

```
crontab -r
```

Description:

Delete all crontab jobs

```
ADD=$(( 1 + 2 ))
echo $ADD
                                    3

MUL=$(( $ADD * 5 ))
echo $MUL
                                   15

SUB=$(( $MUL - 5 ))
echo $SUB
                                   10

DIV=$(( $SUB / 2 ))
echo $DIV
                                    5
MOD=$(( $DIV % 2 ))
echo $MOD

                                    1
```

Command:

```
expr length "This is myw3schools.com"
```

Description:

Find the length of a string (**This is myw3schools.com**)

```
echo '3+5' | bc
                                        8

awk 'BEGIN { a = 6; b = 2; print "(a + b) = ", (a + b) }'

                                    (a + b) =  8
```

Command:

```
factor 10
```

Description:

Decompose an integer **(10)** into prime factors

Command:

```
ps -e
```

Description:

Display every active process on a Linux system

Command:

```
ps -x
```

Description:

Display User Running Processes

Command:

```
ps -fU manju
```

Description:

Display a user's processes by user name (**manju**)

Command:

```
ps -fu 1000
```

Description:

Display a user's processes by real user ID (**RUID**)

Command:

```
ps -U root -u root
```

Description:

Display every process running with root user privileges (real and effective ID)

```
echo -e "The following users are logged on the system:\n\n $(who)"


              manju    :0           Aug 15 03:31 (:0)

              manju    pts/1        Aug 15 03:32 (:0)
```

Command:

```
sh <(curl https://nixos.org/nix/install) --daemon
```

Description:

Install Nix Package Manager in Linux

Command:

```
locale
```

Description:

View System Locale in Linux

Command:

```
locale -a
```

Description:

Display a list of all available locales

```
cat score.txt

                                    Justin-40


cat score.txt | tr [:lower:] [:upper:]

                                    JUSTIN-40


cat score.txt | tr [a-z] [A-Z] >output.txt
cat output.txt

                                    JUSTIN-40
```

```
cat domainnames.txt

                                www. google. com
                                  www. fb. com
                                www. mactech. com

cat domainnames.txt | tr -d ''

                                  www.google.com
                                    www.fb.com
                                  www.mactech.com
```

Remove the spaces in the domain names

```
cat domainnames.txt

                                www.google.....com
                                    www.fb.com
                                  www.mactech.Com

cat domainnames.txt | tr -s ''
                                  www.google.com
                                    www.fb.com
                                  www.mactech.Com

echo "My UID is $UID"

                                   My UID is 0


echo "My UID is $UID" | tr " "  "\n"

                                         My
                                        UID
                                        is
                                         0

echo "myw3schools.com  =>Linux-Books,Src,Tutorials" | tr " " ":"


                    myw3schools.com:=>Linux-Books,Src,Tutorials
```

A space into a " : " character

Command:

```
!sud
```

Description:

Re-execute previously used command

Command:

```
!sudo
```

Description:

Re-execute previously used command

Command:

```
cut -d: -f1 < /etc/passwd | sort | xargs
```

Description:

Generate a compact list of all Linux user accounts on the system

Command:

```
zcat phy.txt.gz myfiles.txt.gz
```

Description:

View multiple compressed files (**phy.txt.gz and myfiles.txt.gz**)

Command:

```
find . -type f -name "*.php"
```

Description:

Find all php files in a directory

Command:

```

```

```
find . -type f -perm 0777 -print
```

Description:

Find all the files whose permissions are 777

Command:

```
find / -type f ! -perm 777
```

Description:

Find all the files without permission 777

Command:

```
find / -perm /g=s
```

Description:

Find all SGID set files

Command:

```
find / -perm /u=r
```

Description:

Find all Read-Only files

Command:

```
find / -perm /a=x
```

Description:

Find all Executable files

Command:

```

```

```
find . -type f -name "phy.txt" -exec rm -f {} \;
```

Description:

Find and remove **phy.txt** File

Command:

```
find . -type f -name "*.txt" -exec rm -f {} \;
```

Description:

To find and remove multiple **.txt** files

Command:

```
find . -type f -name "*.mp3" -exec rm -f {} \;
```

Description:

To find and remove multiple **.mp3** files

Command:

```
find /tmp -type d -empty
```

Description:

Find all Empty Directories

Command:

```
find /tmp -type f -name ".*"
```

Description:

File all Hidden Files

Command:

```

```

```
find / -mtime 50
```

Description:

Find Last 50 Days Modified Files

Command:

```
find / -atime 50
```

Description:

Find Last 50 Days Accessed Files

Command:

```
find / -cmin -60
```

Description:

Find Changed Files in Last 1 Hour

Command:

```
find / -mmin -60
```

Description:

Find Modified Files in Last 1 Hour

Command:

```
find / -amin -60
```

Description:

Find Accessed Files in Last 1 Hour

Command:

Command:

```
type cat
```

Description:

Identifies whether the "cat" command is a shell built-in command, subroutine, alias, or keyword.

```
find / -size 50M
```

Description:

Find all 50MB files

Command:

```
find / -type f -size +100M -exec rm -f {} \;
```

Description:

Find and Delete 100MB Files

Command:

```
find / -type f -name *.mp3 -size +10M -exec rm {} \;
```

Description:

Find all .mp3 files with more than 10MB and delete them

Command:

```
cat /proc/sys/fs/file-max
```

Description:

Find Linux Open File Limit

Command:

```
ulimit -Hn
```

Description:

Check Hard Limit in Linux

Command:

```
ulimit -Sn
```

Description:

Check Soft Limits in Linux

Command:

```
timedatectl  status
```

Description:

Display the current time and date on your system

Command:

```
timedatectl list-timezones
```

Description:

View all available timezones

Command:

```
timedatectl list-timezones |  egrep  -o "Asia/B.*"
timedatectl list-timezones |  egrep  -o "Europe/L.*"
timedatectl list-timezones |  egrep  -o "America/N.*"
```

Description:

Find the local timezone according to your location

Command:

```
timedatectl set-timezone "Asia/Kolkata"
```

Description:

Set your local timezone in Linux

Command:

```
swapon --summary
```

Description:

View a summary of swap space usage by device

Command:

```
cat /proc/swaps
```

Description:

Check swap usage information

```
# start recording of Linux terminal

script history_log.txt

                    Script started, file is history_log.txt

exit

                     Script done, file is history_log.txt
```

Command:

```
dir -shl
```

Description:

List files and their allocated sizes in blocks

Command:

```
less /proc/sys/dev/cdrom/info
```

Description:

Display information about CD-ROM

```
while true; do date >> date.txt ; sleep 5 ; done &

cat date.txt

                        Mon Aug 16 03:05:36 PDT 2021
                        Mon Aug 16 03:05:41 PDT 2021
                        Mon Aug 16 03:05:46 PDT 2021
                        Mon Aug 16 03:05:51 PDT 2021
```

"Don't write better error messages, write code that doesn't need them."

– Jason C. McDonald

```
a=$(pwd)
echo "Current working directory is : $a"
```

```
/home/manju
```

Command:

```
echo *.jpeg
```

Description:

Print all **.jpeg** files

Command:

```
echo 'linux' | fold -w1
```

Description:

Break down a word (**linux**) into individual

```
l
i
n
u
x
```

Command:

```
find . -user root
```

Description:

Output the files with respect of the user **(root)** owned files in the current directory

Command:

```
strace pwd
```

Description:

Trace a command **(pwd)** execution

Command:

```
top -u manju
```

Description:

Display specific User **(manju)** process details

3 characteristics of big data:

- **Volume — How much data is there?**
- **Variety — How diverse is different types of data?**
- **Velocity — At what speed is new data generated?**

Data science process

1: Strategy **(Business understanding)**
2: Data understanding and gathering
3: Data preparation
4: Data exploration
5: Data modeling
6: Evaluation, Presentation and deployment

Command:

```
uname -or
```

Description:

Find Out Linux Kernel Version

Command:

```
uname -a
```

Description:

Print linux system information

Command:

```
cat /proc/version
```

Description:

Display some of your system information including the Linux kernel version

Command:

```
cat /etc/centos-release
```

Description:

Find Out Linux Distribution Name and Release Version

Command:

```
fuser .
```

Description:

Displays the PIDs of processes currently accessing your current working directory

Command:

```
fuser -v -m .bashrc
```

Description:

Determine which processes are accessing your **~.bashrc file**

Command:

```
sudo fuser --list-signals
```

Description:

Displays all the possible signals that can be used with the fuser tool

Command:

```
sudo fuser -k -HUP /boot
```

Description:

Sends the HUP signal to all processes that have your **/boot directory** open

Command:

```
ls -al
```

Description:

List all the files with the file permissions, the number of links to that file, the owner of the file, the group of the file, the file size in bytes, the file's last modified datetime and the file name

Command:

```
echo "shutdown -h now" | at -m 23:55
```

Description:

Shutdown the system at 23:55 today

Command:

```
echo "updatedb" | at -m 23.55
```

Description:

Run **updatedb** at 23:55 today

Creates and updates the database of file names used by locate

 # Everyone can now read the file

chmod a+r myfiles.txt

 # Everyone can now read and write the file

chmod a+rw myfiles.txt

 # Others (not the owner, not in the same group of the file) cannot read, write or execute the file

chmod o-rwx myfiles.txt

Command:

```
echo $(ls -al)
```

Description:

Execute command "ls -al" and print the result to the standard output

Command:

```
top -b -o +%MEM | head -n 22
```

Description:

Display the top 15 processes sorted by memory use in descending order

Command:

```
top -b -o +%MEM | head -n 22 > report.txt
```

Description:

Redirect the output to a file (**report.txt**) for later inspection

Command:

```
ps -eo pid,ppid,cmd,%mem,%cpu --sort=-%mem | head
```

Description:

Check Top Processes sorted by RAM or CPU Usage in Linux

Command:

```
find . -type f \( -name "*.sh" -o -name "*.txt" \)
```

Description:

Find all files in the current directory with **.sh** and **.txt file** extensions

Command:

```
find . -type f \( -name "*.sh" -o -name "*.txt" -o -name "*.c" \)
```

Description:

Find all files in the current directory with **.sh, .c** and **.txt file** extensions

Description:

Find files edited more than 3 days ago.

Command:

```
find . -type f -mtime +3
```

Description:

Find files edited in the last 24 hours.

Command:

```
find . -type f -mtime -1
```

Description:

Find files that have more than 100 characters (bytes) in them.

Command:

```
find . -type f -size +100c
```

Description:

Find files bigger than 100 KB but smaller than 1 MB.

Command:

```
find . -type f -size +100k -size -1M
```

Description:

Deletes all the files edited in the last 24 hours.

Command:

```
find . -type f -mtime -1 -delete
```

Description:

List all files including hidden files.

Command:

```
ls -a
```

Description:

List Files and Directories with "/" Character at the End.

Command:

```
ls -F
```

Description:

List Files in Reverse Order.

Command:

```
ls -r
```

Description:

Sort Files by File Size.

Command:

```
ls -lS
```

Description:

List Files with an inode number.

Command:

```
ls -i
```

Description:

Check the version of the ls command.

Command:

```
ls --version
```

Description:

List files under directory /tmp.

Command:

```
ls -l /tmp
```

Description:

Display UID and GID of files and directories.

Command:

```
ls -n
```

Description:

Find all 30 MB files.

Command:

```
find / -size 30M
```

Description:

Find files with sizes between 100 - 200MB.

Command:

```
find / -size +100M -size -200M
```

Description:

List directories larger than 20 KB.

Command:

```
find / -type d -size +20k
```

Description:

Find empty files and directories.

Command:

```
find ./ -type f -size 0
```

Description:

List files modified within the last 17 hours.

Command:

```
find . -mtime -17 -type f
```

Description:

*** List directories modified within the last 10 days.***

Command:

```
find . -mtime -10 -type d
```

Description:

List all files modified between 6 and 15 days ago in the home directory.

Command:

```
find /home -type f -mtime +6 -mtime -15
```

Description:

Display files with permission 777.

Command:

```
find -perm 777
```

Description:

List files owned by a user (manju).

Command:

```
find /home -user manju
```

Description:

Find all text files owned by user "manju".

Command:

```
find /home -user manju -iname "*.txt"
```

Description:

Find and list files and directories together with their permissions.

Command:

```
find -name "*.conf" | ls -l
```

Description:

List directories only.

Command:

```
ls -d */
```

Description:

List multiple files on a single line.

Command:

```
ls --format=comma
```

Description:

View the process of a specific user "manju".

Command:

```
ps -u manju
```

Description:

Execute a previous command starting with a specific letter "c".

Command:

```
!c
```

Description:

Display BIOS information (You need elevated permissions to run this).

Command:

```
dmidecode -t 0
```

Description:

Display CPU information (You need elevated permissions to run this).

Command:

```
dmidecode -t 4
```

Description:

View all the system logs.

Command:

```
gnome-system-log
```

Description:

Identify SSH Client Version.

Command:

```
ssh -V
```

Description:

Display total connect time of users.

Command:

```
ac -d
```

Description:

Display connect time for all the users.

Command:

```
ac -p
```

Description:

Diaplay connect time report for a specific user "manju".

Command:

```
ac -d manju
```

Description:

Display the modules compiled inside Apache.

Command:

```
httpd -l
```

Description:

*** View Processes Owned by Current User.***

Command:

```
ps U $USER
```

Description:

Display the information about the filesystem Type.

Command:

```
df -Tha
```

Description:

Display Active Connections with Process ID and Program Name.

Command:

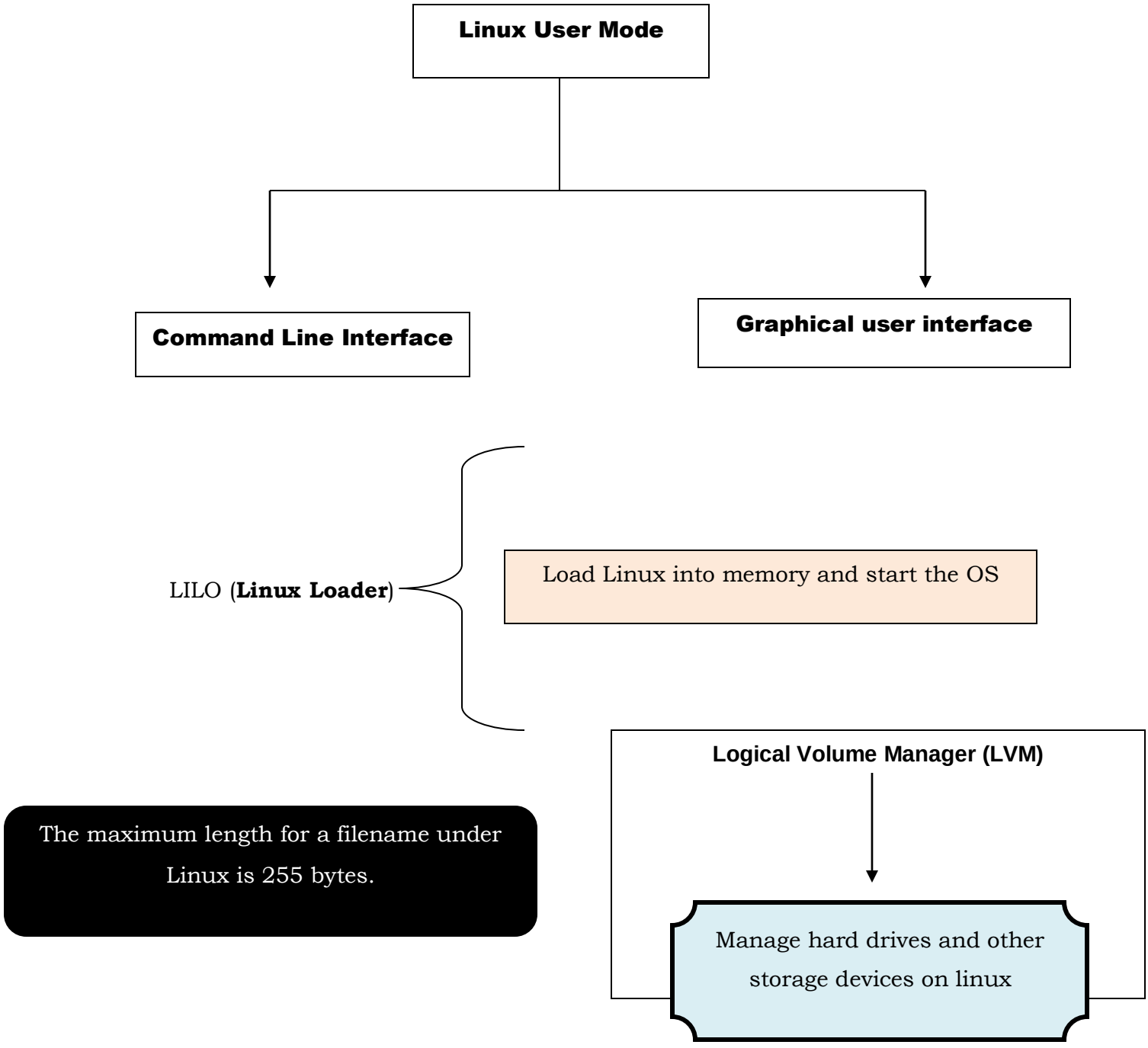
```
netstat -tap
```

Description:

Display RAW network statistics.

Command:

```
netstat --statistics --raw
```


Linux User Mode
Command Line Interface
Graphical user interface
LILO (Linux Loader)
Load Linux into memory and start the OS
Logical Volume Manager (LVM)
Manage hard drives and other storage devices on linux
The maximum length for a filename under Linux is 255 bytes.

References:

- Why learn Linux? By Candace Hazlett
- 11 Reasons Why Linux Is Better Than Windows By Ankush Das
- Linux for Dummies By Richard Blum
- Linux: The Complete Reference By Richard Petersen

"Sladkey recalls the first time he found and sent a bug to Linus: "My first contribution was in porting some program, probably one of my smaller personal projects. I discovered a bug. Since Linux came with source, my first inclination as a hacker was to take a look under the hood and see if I could fix the problem. I found that although I had never done any kernel work, that I was able to navigate around the code pretty easily and provide a small patch to correct the problem. "With my heart beating and my palms sweating, I composed the most professional message I could muster and sent it off to linus.torvalds@cs.helsinki.fi describing the bug and including my proposed fix. Minutes later he replied something like, 'Yup, that's a bug. Nice investigation. Thanks. Fixed,' and I was hooked."

— Glyn Moody